An on-going framework for easily experimenting with deep learning models for bioimaging analysis*

Manuel García, César Domínguez, Jónathan Heras, Eloy Mata and Vico Pascual

Dpto. de Matemáticas y Computación, Universidad de La Rioja
Edificio CCT, Madre de Dios, 53, E-26006 Logroño (La Rioja, Spain)
{manuel.garcia, cesar.dominguez, jonathan.heras, eloy.mata, vico.pascual}@unirioja.es

Abstract. Due to the broad use of deep learning methods in Bioimaging, it seems convenient to create a framework that facilitates the task of analysing different models and selecting the best one to solve each particular problem. In this work-in-progress, we are developing a Python framework to deal with such a task in the case of bioimage classification. Namely, the purpose of the framework is to automate and facilitate the process of choosing the best combination of feature extractors (obtained from transfer learning and other techniques), and classification models. The features and models to test are fixed by a simple configuration file to facilitate the use of the framework by non-expert users. The best model is automatically selected through a statistical study, and then it can be employed to predict the category of new images.

Keywords: Deep Learning, Machine Learning, Parallelization, Bioimaging, Image Processing

1 Problem statement

Nowadays, there exists an increment in the use of deep learning methods in a wide variety of computer vision applications. These methods can be used together with more traditional techniques in image processing. There is not a silver bullet solution to solve the problem. In addition, these methods can be successfully applied in areas such as Bioimaging [1–3] where the specialists could not have the knowledge to use them. Therefore, these techniques are beyond the possibilities of some of the researches. The aim of this work consists in trying to facilitate the use of these methods to non-expert users in object bioimage classification problems.

* This work was partially supported by Ministerio de Economía, Industria y Competitividad [MTM2014-54151-P, MTM2017-88804-P], and Agencia de Desarrollo Económico de La Rioja [ADER-2017-I-IDD-00018].

2 Related work

This project can be framed in the context of democratising artificial intelligence and more particularly deep learning techniques. In this line, we can highlight two projects. The first one [4] compares five different model classifiers with your dataset; and the model with the best accuracy is used to carry out the next comparisons. The user provides the dataset as an input and get automatically created a high-performance ready-to-use pattern recognition system. The second one [5] helps users to create their own classifiers from a two-dimensional visual interface. This project guides to non-experts users to create their own models without any help of a learning algorithm. This kind of users can get good classifiers using that simple user interface. Expert users can also use the tool to produce the models. They can build models with better results than those generated by a learning algorithm and thus these models be more intelligible. Both of them compare the clasiffiers with the dataset that you want to study and give us the classifier with the best accuracy.

Other related framework has just launched by Google. This platform is called Google Cloud AutoML [6]. It helps not-expert users to generate automatic learning models in a simple way. At the moment it is in a testing phase and you need permission to have access.

All these projects have some drawbacks. For instance, they focus only on deep learning models. They are also not extensible in the sense that they do not allow user to add new models.

3 Proposal

We propose a framework that tries to solve some of the previous drawbacks. Figure 1 includes the workflow of the intended framework proposal. It tries to facilitate the task of defining models for bioimage classification using classical image processing and deep learning techniques (using transfer learning with neural networks in Keras such as VGG16, VGG19, Resnet, Inception, etc). The workflow can be divided in four different phases: dataset and configuration file, feature extraction, statistical analysis and training the best model and predicting. We are going to detail such as phases in the following subsections.

3.1 Dataset and configuration file

The input of the framework is a dataset of images and a configuration file. The dataset consists of folders of images where each folder represents a category. In addition, a configuration file containing the parameters of the program is necessary. We have to specify the path of the dataset, a list of feature extractors and a list of classification models.

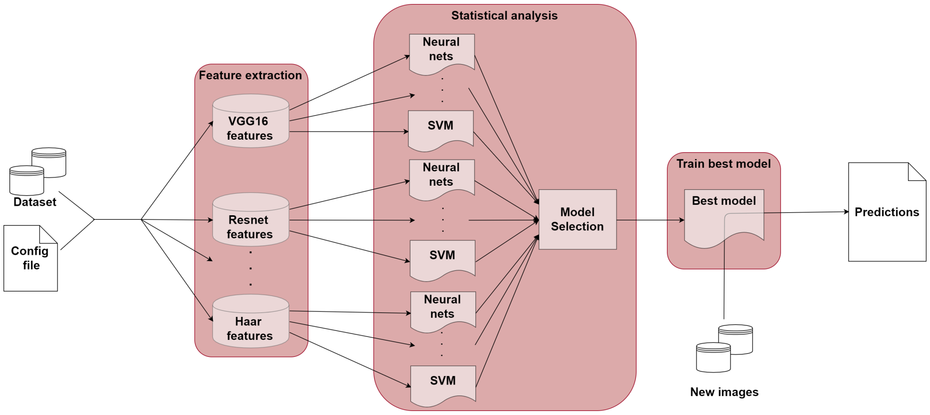

Fig. 1. Workflow of the framework proposal

3.2 Feature extraction

In this step, the framework collects the features of each image of the dataset using both transfer learning and classical computer vision techniques. For each feature extractor provided in the configuration file, the framework goes through each image and extracts the features. Afterwards, this information is saved in an h5py file for later use.

3.3 Statistical analysis

Based on the features gathered in the previous step, the framework carries out a statistical study [7], to compare several classification models and obtain the best one. This study measures the accuracy of each combination of a feature extractor and a classification model using cross-validation. Namely, for each extractor, the best classification model is obtained. Subsequently, those models are compared to identify the best combination of feature extractor and classification model.

3.4 Training the best model and predicting

In the last step, the best featureExtractor-modelClassifier combination is trained. The framework trains the model with all the dataset images.

Once the model is trained, it is exported to a file that can be used later to predict the classes of the images that are passed to the model. The whole process is automatically performed by the framework.

4 Preliminary Results

At this moment the framework is under development using Python. It is used OpenCV [8] to extract classification features using classical image techniques and Keras [9] to extract classification features using deep learning. Besides, a combination of these features extracted with several classification models is performed; and the best one is selected using a statistical analysis based on the accuracy of the models. In the near future, a full test on a real dataset will be carried out to study the suitability of the proposal. Since the techniques including in these processes are very time consuming, we could apply parallelization techniques.

5 Reflections

As we can see, this project tries to bring artificial intelligence closer to those users without notions about what a deep learning model is or how to configure the models correctly to obtain good predictions. The aim consists in trying to bring this powerful technology to most people so they can easily make use of its benefits.

It is expected that the best model will be incorporated automatically into a web application that will guide those non-expert users in their bioimage classification problems.

Apart from the above, it is planned to look for new deep learning libraries to increase the number of models. In addition, a new option will be added to the configuration file to choose different measures (beyond accuracy) to calculate the performance of the models so that we can improve the user's options and select the model that best suits the problem in each case.

References

1. Ronneberger, O., Fischer, P., Brox, T.: U-Net: Convolutional Networks for Biomedical Image Segmentation (2015), http://arxiv.org/abs/1505.04597
2. Akkus, Z., Galimzianova, A., Hoogi, A., Rubin, D.L., Erickson, B.J.: Deep Learning for Brain MRI Segmentation: State of the Art and Future Directions. Journal of Digital Imaging 30(4), 449–459 (2017)
3. Nauman, M., Ur Rehman, H., Politano, G., Benso, A.: Beyond Homology Transfer: Deep Learning for Automated Annotation of Proteins. bioRxiv (2017), https://www.biorxiv.org/content/early/2017/07/25/168120
4. Reif, M., Shafait, F., Goldstein, M., Breuel, T., Dengel, A.: Automatic classifier selection for non-experts. Pattern Analysis and Applications 17(1), 83–96 (2012)
5. Ware, M., Frank, E., Holmes, G., Hall, M., Witten, I.H.: Interactive machine learning: letting users build classifiers. International Journal of Human-Computer Studies 55(3), 281–292 (2001)
6. Google: Google cloud automl (2018), https://cloud.google.com/automl/
7. Sheskin, D.: Handbook of Parametric and Nonparametric Statistical Procedures. CRC Press (2011)
8. Intel: Opencv (2016), https://opencv.org/
9. F. Chollet, e.a.: Keras (2017), https://keras.io/